An Application of Hierarchical Gaussian Processes to the Detection of Anomalies in Star Light Curves

Niall Twomey[a,*], Haoyan Chen[a], Tom Diethe[a,b,b], Peter Flach[a]

[a]*Intelligent Systems Laboratory, University of Bristol UK*
[b]*Amazon Research, Cambridge, UK*

Abstract

This study is concerned with astronomical time-series called *light-curves*, that represent the brightness of celestial objects over a period of time. We consider the task of finding anomalous light-curves of periodic variable stars. We employ a Hierarchical Gaussian Process to create a general and stable model of time-series for anomaly detection, and apply this approach to the light-curve problem. Hierarchical Gaussian Processes require only a few additional parameters compared to conventional Gaussian Processes and incur negligible additional computational complexity. Moreover, since the additional parameters are objectively optimised in a principled probabilistic framework one does not need to resort to grid searches for parameter selection. Experimentally, we demonstrate that our approach outperforms several baselines on both synthetic and light-curve data. Of particular interest is that the proposed method generalises very well from small subsets of the data, achieving near perfect precision of outlier detection even with as few as seven instances.

Keywords: astronomical data, anomaly detection, Gaussian processes

1. Introduction

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour; we typically call these non-conforming patterns anomalies, outliers, aberrations, exceptions, etc. [1]. Our target is to find periodic anomalous time-series (rather than anomalies within the time-series itself) using unsupervised learning approaches. One of the major tasks in astronomy is to detect when aberrant phenomena are encountered from historical observations [2], and it is almost impossible to find the anomalous objects through manual inspection due to the scale of the data.

In astronomy, light-curves are real-valued time-series of light magnitude measurements that show the brightness of a celestial object or region. The study of periodic variable

*Corresponding author

Email addresses: niall.twomey@bristol.ac.uk (Niall Twomey), capechy@hotmail.com (Haoyan Chen), tom.diethe@amazon.com (Tom Diethe), peter.flach@bristol.ac.uk (Peter Flach)

[1]NT is funded by Medical Research Council Momentum Awards under Grant MC/PC/16029.

[2]Work done prior to joining Amazon.

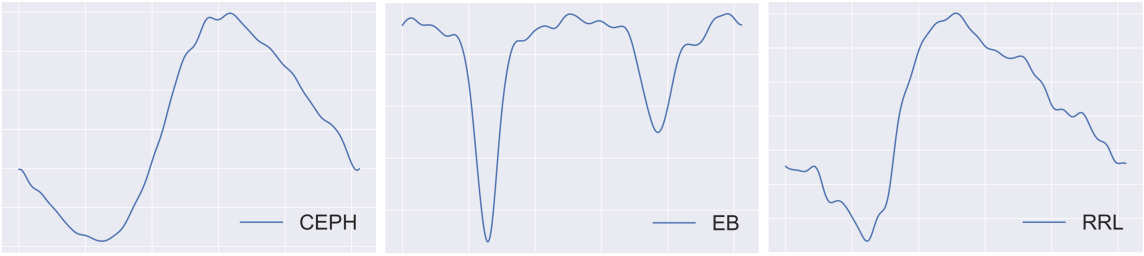


Figure 1: Example of one period of three light-curves: Cephoid (left), eclipsing binary (middle) and RR Lyrae (right). Notice that the CEPH and RRL light-curves depict similar patterns, whereas the EB curves are quite distinct.

stars is an important factor in astronomy. For example, Cephoid (CEPH) variable stars can be a means to calibrate the extragalactic distance scale, which plays a crucial role in launching and repairing of the Hubble Space Telescope [3]. Finding new periodic variable stars and improving the efficiency of automatic detection of existing start classes are both of significant value to astronomy. Figure 1 shows a typical light-curve from the Optical Gravitational Lensing Experiment (OGLE) [4] for periodic variable stars after data pre-processing and re-sampling [5]. Cepheid, Eclipsing Binaries (EB) and RR Lyrae (RRL) are common types of periodic variable stars, and the details of these three stars can be found in [6]. The three light-curves in Figure 1 are typical, but the real light magnitude measurements for each star may have been incorrectly classified. This means that the dataset may contain an unknown number of additional outliers. Our goal is to introduce a robust approach to detecting these outliers.

Probabilistic models making use of Gaussian Processes (GPs) [7] have become a standard approach to solving a variety of machine learning problems, due to their flexibility, quantification of uncertainty, and calibrated probabilistic outputs. GPs have a had a long history in time-series analysis [8] in a variety of fields. Hierarchical GPs (HGPs) [9] are a recent method for tasks involving multiple related time-series, in which there is assumed to be some underlying generative process shared between the observations. HGPs were originally proposed for the analysis of gene expression data, where the multiple time-series are seen as noisy realisations of a common underlying driver process that is exhibited during the synthesis of a functional gene product (e.g. a protein). Our central hypothesis is that this intuition is also a fundamental property of light-curves, where the underlying function represents the periodic physical variation of the celestial bodies.

The remainder of the paper takes the following structure. In Section 2 we review related work. We review anomaly detection in general and then specifically we review anomaly detection with time-series data. Section 3 then introduces the mathematical foundations of Gaussian processes (GPs). This section also introduces the hierarchical formulation for GPs and how they are utilised in our work to produce anomaly scores. Section 4 introduces the datasets that we use for our analysis as well as two baseline methods, and Section 5 summarises the associated results. Finally, we conclude in Section 6.

2. Probabilistic Anomaly Detection

Time-series anomaly detection is the focus of study in several domains including time-series clustering, anomaly detection *within* time-series, and identification of anomalous *whole* time-series. We focus our attention to the latter two items here. In particular we consider models that are probabilistic in nature, first concerning ourselves with parametric approaches of density estimation and concluding with non-parametric techniques since this is the setting of the proposed model. Not all anomaly detection techniques necessarily fall in the remit of probabilistic models (including dynamic time warping [10], neural networks [11], grammar mining [12], max-margin methods [13]) and the interested reader is referred to [14, 15] for overviews of such methods. In several of the cases below we will see that measures of likelihood are a suitable surrogate for anomaly estimation, an idea which our approach also incorporates.

Estimating the generative process of the data is the key starting point for anomaly detection in the approaches outlined here. Of course, this estimation process is non-trivial for several reasons: the true generative process is latent and many variables may contribute to this via non-linear pathways; also, only a finite sample of data is available for parameter inference. The latter point in particular can lead to under- and over-fitting when there is a mismatch between the true generative model and its estimate [16]. Typically, and in particular with the absence of data, model parsimony is deemed preferable to highly complex alternatives [17, 18], an idea often called 'Occam's razor' of model selection. We will return to this point later in subsequent sections.

A classic category of density models are finite mixture models, and in particular Gaussian Mixture Models (GMMs) [19]. GMMs are a generative model wherein data are assumed to be generated from $K \in \mathbb{N}$ weighted (multivariate) Gaussian distributions as follows

$$p(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where $\mathbf{x_i}$ is the i-th instance of a dataset, $\mathcal{N}(\cdot)$ is the Gaussian distribution, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of the k-th mixture, π_k is the K-vector of mixture coefficients (satisfying $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0 \; \forall \; 1 \leq k \leq K$), and θ is the set containing all means, covariances and mixing coefficients. It is easy to see that such a model can capture more complex densities than any single distribution, for example, since it can cover a larger class of (multi-modal) distributions. We can see the value of K therefore as controlling the complexity of the model since larger values of K will induce a larger number of parameters. However, one must be wary of arbitrarily increasing K since models will tend to over-fit to the training data as K grows [19, 20]. Hence, we can see that on one hand there are advantages in giving models sufficient capacity to capture the dynamics of the data, but on the other hand the flexibility of the models must be moderated to ensure that they generalise to unseen data. These models have well-defined likelihood functions which may be used for measuring the degree to which an instance is an inlier or outlier. This basic principle, in

particular when combined with extreme value theory [21], has been applied successfully in several application areas from healthcare [22], activity recognition [23], anomaly detection with natural gas [24] and speaker verification [25]. A criticism of GMMs is that although they are a flexible model of data they are unlikely to represent the true dynamics of the domain problem. This is a key challenge for all models but we introduce several approaches below that can capture more complex examples of the richness of data.

An exciting recent trend in probabilistic machine learning has been to imbue probabilistic models explicitly with domain knowledge [26], *e.g.* with factored relationships between random variables. In so doing, model inference can be expedited, model interpretability can be increased and models can be fairly compared for model selection since bespoke architectures have been crafted for the problem [26]. It is often non-trivial to encode this domain knowledge into the language of graphical models, but when successful this can produce elegant and accurate models in many domain areas, including skill assessment [27], matching [26], reviewing [28] and recommendation systems [29]. Indeed, this is the key idea that has recently inspired a revolution in probabilistic programming frameworks in the machine learning field [30, 31, 32, 33] that offer a high-level interface for modelling and factorising data. Although GMMs are naturally expressible as Bayesian networks, more general and arbitrary graphical structures can be imposed in modelling, *e.g.* recently [34] demonstrated that complex Bayesian networks can be applied to capture anomalies in maritime application areas in time-series for the purpose of security. This work involved the analysis of many variables and the complexity of the problem necessitated densely connected network structures to adequately detect anomalies. More generally, dynamic networks deal with graphs that evolve over time or sequential data [20] and these have also been utilised with great success in detecting anomalous time-series [34]. However, these models are often underpinned by low-order Markovian assumptions regarding the graph dynamics which may be incapable of capturing the true dynamics of the time-series [35].

Until now the capacity and complexity of the models used for anomaly detection have been specified in advance in a parametric manner, *i.e.* a practitioner defines the number of parameters of the models by specifying K or by defining the connectivity of the Bayesian network. Some work, however, has been done to increase the capacity of anomaly models with hierarchical [36] and ensemble [37] anomaly techniques. However, these methods are still parametric. On the other hand, non-parametric models are a flexible model class in which the capacity of the model is permitted to 'grow' with data. Hence, if a non-parametric model is applied to a small dataset, 'simple' posterior distributions may be inferred while more 'complex' models may arise with more data. Indeed some of the models discussed already can be generalised to so-called 'infinite' variants. For example, the Dirichlet Process GMM (DPGMM) [38] and Infinite Hidden Markov Model (IHMM) [39] are infinite examples of the GMM and Hidden Markov Model (HMM) (a dynamic Bayesian network) where explicit specification of model complexity is treated as another latent variable and hence is inferred from the data.

It is in the non-parametric family of models that our proposed approach, which is rooted in Gaussian Processes (GPs) [7], is based. The key difference between the methods described previously and GPs is that the explicit graphical structure of the previous models is forsaken

in favour of covariance functions that measure similarity between instance pairs. We will show later that this approach reduces inference of GPs to the arithmetic of multivariate Gaussians. The practitioner's role then evolves from specifying a graphical structure between all variables to that of measuring similarity/covariance, which in turn corresponds to encoding prior beliefs about the forms of the functions being represented, such as smoothness or periodicity.

Users of GPs have also benefited from the recent abstractions achieved with probabilistic programming interfaces – notable examples include [40, 41] – and this has contributed to significant interest in these methods. In our work GPs are used for anomaly detection. This idea has been used in other application areas and in [42] the authors utilise GPs to detect and correct for anomalies in sensor data (*e.g.* correcting drift bias) but they demonstrate excellent performance, even with small volumes of data. An interesting challenge in their setting is that non-stationary covariance functions were required since the probability of the data changes dramatically due to faults. Other successes of GPs for anomaly detection includes [43] where GPs are used in non-*iid* settings. This is the scenario that we we experience in time-series data analysis since neighbouring time points will be highly correlated. We consider start-light-curves in this work, which are measures of the 'brightness' of stars. These processes are relatively stable and unlikely to elicit significant variation in orbits like the above, so there is no need to consider non-stationary covariance functions. Even though these data are well-behaved they will elicit complex time-series patterns that are received through noisy media (*e.g.* weather will affects luminosity). Both of these challenges can naturally be handled by GPs and hence are considered as the modelling framework of choice here. The majority of approaches in machine learning applied to star light-curves, apart from our earlier work [44], falls outside the Bayesian paradigm. Although these methods have been successful, our analysis provides strong evidence for the benefits associated with non-parametric probabilistic models in these domains, owing particularly to their ability to quantify uncertainty, utilise data efficiency, and model selection.

3. Gaussian Processes and Proposed Approach

This work builds up directly from our earlier work in astroinformatics in star light-curve anomaly detection [44]. We will briefly introduce the process of GP regression in a Bayesian framework before then discussing hierarchical variants of this model. We then conclude this section with the proposed 'anomaly score' that is utilised in our empirical evaluation.

3.1. Gaussian Processes

Here we are analysing time-series, which will be denoted by a set of time points $\mathbf{x} = \{x_t\}_{t=1}^T$ with their respective responses (*i.e.* light-curve intensities) $\mathbf{y} = \{y_t\}_{t=1}^T$. We cast time-series analysis into a regression problem of the form $y = f(x)+\eta$ where η is independent Gaussian noise with precision τ. Our goals will be to estimate a functional form of f and to evaluate the distribution of y^* given a particular x^*, *i.e.* $p(y^*|x^*)$. A GP is a distribution over functions, and is specified by its mean function $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$ that quantifies the covariance between input points $\mathbf{x}$, compactly $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^\top))$,

and $p(\mathbf{y}|f(\mathbf{x})) = \mathcal{N}(f(\mathbf{x}), \tau^{-1}\mathbf{I}_T)$ where $\mathbf{I}_T$ is the $T \times T$ identity matrix and τ is the noise variance.

The choices of mean and covariance functions are based on an understanding of the domain. The mean function allows use to impose our prior beliefs on the behaviour of the functions far away from observed data. Here, as is often the case, we assume that the trend is to return to zero, which corresponds to the mean function $\mu(x) = 0$. In the case of light-curves we need covariance functions that can express both smooth variation and small fluctuations (see Figure 1). As a practitioner, one has the important task of selection the covariance functions for the task at hand. Several standard functions are often used, including linear, Radial Basis Function (RBF), and Matern[3], and these are expressed below for scalar inputs.

$$k_{\text{LIN}}(x, z) = \sigma^2 (x \cdot z), \tag{2}$$

$$k_{\text{RBF}}(x, z) = \sigma^2 \exp\left(-\frac{(x-z)^2}{\ell}\right), \tag{3}$$

$$k_{\text{MAT32}}(x, z) = \sigma^2 \left(1 + \frac{\sqrt{3}\,|x-z|}{\ell}\right) \exp\left(-\frac{\sqrt{3}\,|x-z|}{\ell}\right), \tag{4}$$

where ℓ, σ are the hyperparameters of the covariance functions. Although we show three covariance functions in isolation, these can be composed together by multiplication and addition [45]. Such compositions allow practitioners to impose domain knowledge into the model (*e.g.* linear and periodic covariance functions may be constructed where these correlations are present in data).

Given our training time-series $\mathbf{x}$ as defined above, and test instances $\mathbf{x}^* \in \mathcal{R}^{T^*}$ defined similarly for a set of testing time points of length T^*, due to the properties of GPs [7] we have that the train and test data are jointly Gaussian as follows:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \boldsymbol{\mu}_{\mathbf{x}^*} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{xx}} & \mathbf{K}_{\mathbf{xx}^*} \\ \mathbf{K}_{\mathbf{x}^*\mathbf{x}} & \mathbf{K}_{\mathbf{x}^*\mathbf{x}^*} \end{bmatrix}\right), \tag{5}$$

where $\mathbf{y}$, $\mathbf{y}^*$ are the training responses and the test predictions, $\boldsymbol{\mu}_{\mathbf{x}} = \mu(\mathbf{x})$, $\boldsymbol{\mu}_{\mathbf{x}^*} = \mu(\mathbf{x}^*)$ are the means for train and test examples respectively, and $\mathbf{K}_{\mathbf{xx}} = k(\mathbf{x}, \mathbf{x}^\top)$, $\mathbf{K}_{\mathbf{x}^*\mathbf{x}^*} = k(\mathbf{x}^*, \mathbf{x}^{*\top})$, $\mathbf{K}_{\mathbf{x}^*\mathbf{x}} = k(\mathbf{x}^*, \mathbf{x}^\top)$ and $\mathbf{K}_{\mathbf{xx}^*} = k(\mathbf{x}, \mathbf{x}^{*\top})$ denote the train, test, test-to-train and train-to-test covariance sub-matrices whose elements are derived using the covariance function k. Since $\mathbf{y}$ (the training labels) are given, the posterior distribution of $\mathbf{y}^*$ (test predictions) can be calculated simply by conditioning on the training data [7] and predictions follow a Gaussian as follows:

[3]Here we present the Matern 3/2 kernel. A more general form can be found in [7].

$$
\begin{aligned}
p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) &= \mathcal{N}(\mu_{\mathcal{D}}(\mathbf{x}^*), k_{\mathcal{D}}(\mathbf{x}^*, \mathbf{x}^*)), \\
\mu_{\mathcal{D}}(\mathbf{x}^*) &= \boldsymbol{\mu}_{\mathbf{x}^*} + \mathbf{K}_{\mathbf{x}^*\mathbf{x}}\mathbf{K}_{\mathbf{xx}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}}) \\
k_{\mathcal{D}}(\mathbf{x}^*, \mathbf{x}^*) &= \mathbf{K}_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*\mathbf{x}}\mathbf{K}_{\mathbf{xx}}^{-1}\mathbf{K}_{\mathbf{x}\mathbf{x}^*},
\end{aligned} \tag{6}
$$

where $m_{\mathcal{D}}$ and $k_{\mathcal{D}}$ are the posterior mean and covariance of the predictive distribution. Notice that the posterior variance $k_{\mathcal{D}}(\mathbf{x}^*, \mathbf{x}^*)$ is always smaller than the prior variance $\mathbf{K}_{\mathbf{x}^*\mathbf{x}^*}$ because $\mathbf{K}_{\mathbf{x}^*\mathbf{x}}\mathbf{K}_{\mathbf{xx}}^{-1}\mathbf{K}_{\mathbf{x}\mathbf{x}^*}$ is always positive.

The covariance functions hyperparameters will be selected using Type-II maximum likelihood [7]. We first express the log marginal likelihood of the training data below. Note that this expression is conditioned on θ which is a set containing the hyperparameters of the mean and covariance functions (*e.g.* for the RBF kernel, $\theta = \{\sigma, \ell\}$).

$$
L = \log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\log|\mathbf{K}_{\mathbf{xx}}| - \frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})^\top \mathbf{K}_{\mathbf{xx}}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}) - \frac{T}{2}\log(2\pi) \tag{7}
$$

Then, we calculate the gradient of this with respect to all hyperparameters as follows [7]:

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_\mu} &= -(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})^\top \mathbf{K}_{\mathbf{xx}}^{-1}\frac{\partial \boldsymbol{\mu}_{\mathbf{x}}}{\partial \theta_\mu} \\
\frac{\partial L}{\partial \theta_k} &= \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\mathbf{xx}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{xx}}}{\partial \theta_k}\right) + \frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})^\top \frac{\partial \mathbf{K}_{\mathbf{xx}}}{\partial \theta_k}\mathbf{K}_{\mathbf{xx}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{xx}}}{\partial \theta_k}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})
\end{aligned} \tag{8}
$$

where θ_μ and θ_k are used to denote the subsets of hyperparameters on the mean and covariance function respectively, and $\mathrm{tr}(\cdot)$ is the matrix trace operator. Equation (8) provides an expression of the gradient of the marginal likelihood with respect to the hyperparameters of the model. In this work we do not specify a range of hyperparameter values (*e.g.* with grid search) but instead follow the analytic gradient of the marginal likelihood until convergence has been reached in Equation (8), *i.e.* the hyperparameters are learnt during inference. A convenient byproduct of Type-II maximum likelihood is that the marginal likelihood can be used effectively for model selection, although inference may need to be repeated since Equation (7) is not convex with respect to θ. Traditional parameter selection in machine learning (*e.g.* via grid search) is more restrictive since only parameters within the pre-specified set are considered, whereas with Type-II maximum likelihood the parameters adapt according to the gradient in Equation (8).

Although we optimise model hyperparameters during inference, we can still study the effect of hyperparameter configurations on randomly sampled data. Figure 2 shows five draws from a GP prior with RBF kernel with 9 different configurations of covariance function hyperparameters. As σ decreases from top to bottom, we can observe that the scale of the samples likewise drops. Similarly, as ℓ increases from left to right, we can see that the length-scale of the sampled functions increases where we see the highest frequency signals on the left, and the lowest on the right. Thus, we can understand that the length-scale ℓ determines the length of 'wiggles' and the variance σ determines the amplitude of the functions [46].

Note also that we show only samples from the prior distribution here. Samples from posterior models will produce functions that pass through (or near) the training data.

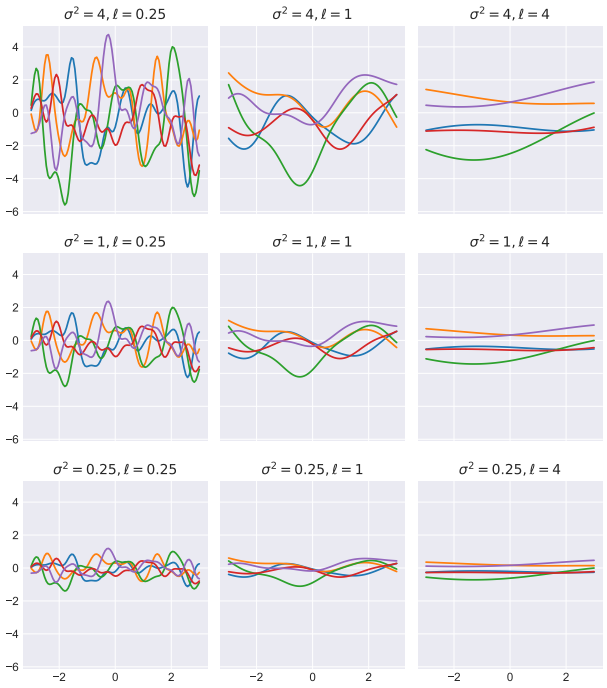


Figure 2: Samples from a prior GP using Radial Basis Function (RBF) kernel. In each column of this figure, the shape of the functions are same, but they have increasing amplitudes (average distance away from the mean line $y = 0$) as σ increases. Meanwhile, each row depicts a different values for the parameter ℓ that determines the length-scale.

3.2. A hierarchy across time-series

The key idea underpinning Hierarchical GPs (HGPs) for star light-curves is that all star light-curves of the same type have a shared underlying function (see star light-curve examples in Figure 1). However, individual stars will exhibit individual variation from these prototypical curves, sometimes due to environmental factors (*e.g.* weather conditions during recording) and sometimes due to the properties of the individual star (*e.g.* the 'brightness' of the star will introduce some variation). For the purposes of the present paper these factors are considered as noise since we are only interested in capturing the dynamics of the underlying periodic curves in order to detect anomalies. HGPs are a probabilistic model that can capture this tiered structure by factorising the covariance matrix in a hierarchical manner [9], and our suggestion is that this hierarchical Bayesian structure is an excellent fit for describing the light-curves of periodic variable stars.

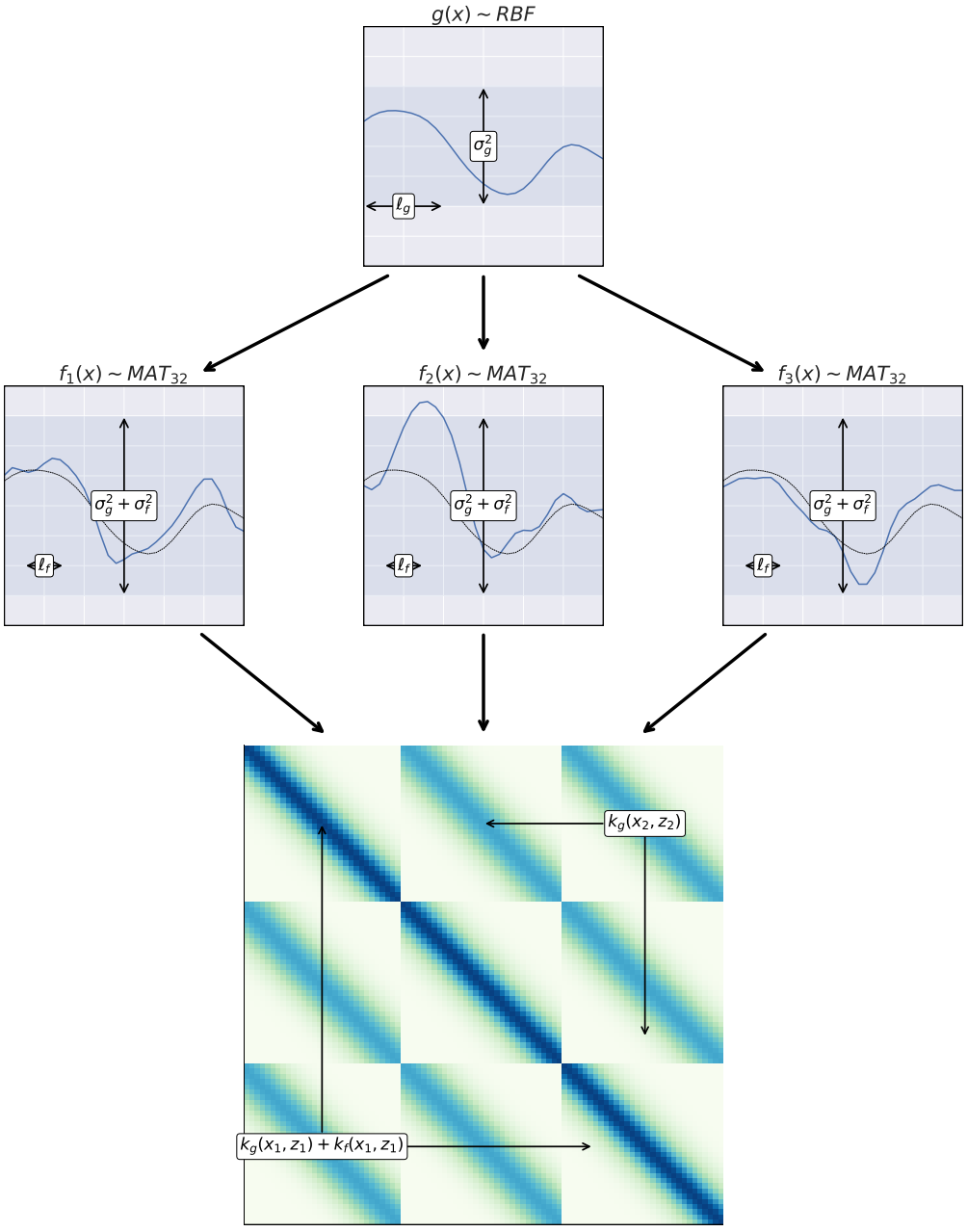


Figure 3: On top, the underlying function, g, is shown, with a smooth behaviour. Each of the replicates in the middle images (f_1, f_2, f_3) are draws from this (note the background function is also shown in the thin black trace). Although the main aspects of the underlying function are preserved, each of the replicates has its own individual characteristics. Finally, the bottom image demonstrates the manner in which covariance matrices are augmented and composed together, and four instances (two along the leading block diagonal, and two off this diagonal) are highlighted.

Figure 3 shows an example of draws from a HGP. This image shows a basic underlying function (top) and several samples from the distribution over this (middle) and how the

data are composed into the covariance matrix (bottom). What makes HGPs powerful is that each replicate may have different noise characteristics, and in this way the model has significant capacity to model a wide range of variants of the basic underlying function.

As we will show later, the HGPs used in this work are composed of a two-level hierarchy, although deeper hierarchies could be modelled. The first covariance function is used to capture the underlying profile of the star light-curves and the second covariance function captures the individual variance of the 'replicate' stars [9]. The HGP model is constructed as follows:

$$\begin{aligned} g(\mathbf{x}) &\sim \mathcal{GP}(\mathbf{0}, k_g(\mathbf{x}, \mathbf{x}^\top)) \\ f_i(\mathbf{x}) &\sim \mathcal{GP}(g(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}^\top)) \quad i = 1, \ldots, m \end{aligned} \tag{9}$$

where g is the underlying function, f_i is the function for the i^{th} replicate function, $k_g(\cdot, \cdot)$ and $k_f(\cdot, \cdot)$ are the covariance functions of the underlying and replicate GPs respectively. The two covariance functions are then selected to reflect beliefs about the nature of the true underlying function and how replicate functions vary from this underlying function.

A dataset will contain n replicates of the time-series for a given star type, and we concatenate the time points and responses into the variables $\hat{\mathbf{x}} \in \mathcal{R}^{nT}$ and $\hat{\mathbf{y}} \in \mathcal{R}^{nT}$. Owing to the conjugacy of Gaussians, given training data $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ the the model above can be expressed as a jointly Gaussian distribution as follows:

$$p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \theta) \sim \mathcal{N}\left(\mathbf{0}, k_n\left(\hat{\mathbf{x}}, \hat{\mathbf{x}}^\top\right)\right) \tag{10}$$

where

$$k_n(x, z) = \begin{cases} k_f(x, z) + k_g(x, z) + \tau^{-1} & \text{if } x \text{ and } z \text{ are from the same replicate} \\ k_f(x, z) & \text{otherwise.} \end{cases} \tag{11}$$

and τ^{-1} is the noise variance. We can see the structure achieved by this in Figure 3 (bottom). Notice the block structure between replicates and the influence of within-replicate covariance on the leading diagonal. Posterior inference and optimisation of covariance parameters is done with the same techniques described in Section 3.1. Indeed, one of the elegant features of this model is that even though the GP has been imbued with additional capacity the inference still follows the traditional methods described in Section 3.1.

Since the light-curves exhibit both low frequency oscillations and high frequency noise, we have selected the Matern kernel for k_f and k_g. Figure 4 illustrates how the HGP model we build performs on the three types of astronomical stars. The left-most column defines the underlying latent functions inferred by the HGP, and the remaining subplots show example light-curves from the OGLE dataset.

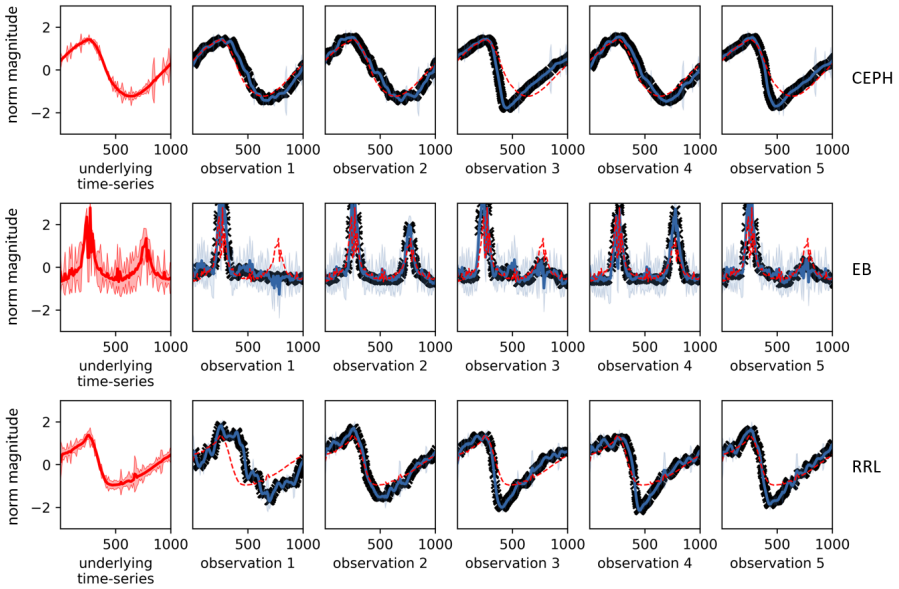


Figure 4: HGP on the three star types. The leftmost sub-figures represent the function $f_n(t)$, and the subsequent figures show five different measurements. The mean (and 95% confidence interval) are shown in blue (shaded), and the the mean of underlying function $f_n(t)$ in red.

3.3. Modelling Periodicity

A natural question arises, since we know that these are time-series from periodic events, as to whether we can explicitly model the periodicity of the signal. There are classes of kernel functions that directly model periodicity, such as the periodic extensions of the RBF or Matern kernels (see [7]). However, we note that whilst the periodicity is strong, we are (roughly) only observing a single period within a single observation (time-series). We show the effect of using a periodic Matern 3/2 (summed with an RBF kernel with a long length-scale, to capture the 'tilted' nature of the periodic function) base kernel, whilst using the Matern 3/2 as the replicate kernel as before. We can see two examples of this in Figure 5, where we can see that in the upper figure, the model has found a period of around 22, corresponding to local quasi-periodic fluctuations in the signal. We also note that the log marginal likelihood of this is around -1537, which is much smaller than the -1287 that we achieve using a standard Matern kernel, indicating that the latter would be the preferred model to select.

After many (tens of) optimization restarts, we were able to find the solution in the lower figure, where the correct period has been found (with a log marginal likelihood of -1210). However, firstly this solution is extremely unstable - in general the optimization procedure does not converge on this solution; and secondly, in some sense this has 'overfit', in the sense that the error bars around the base kernel look overly tight. For the rest of the paper, we chose instead to focus on the more stable solutions provided by the non-periodic kernel functions.

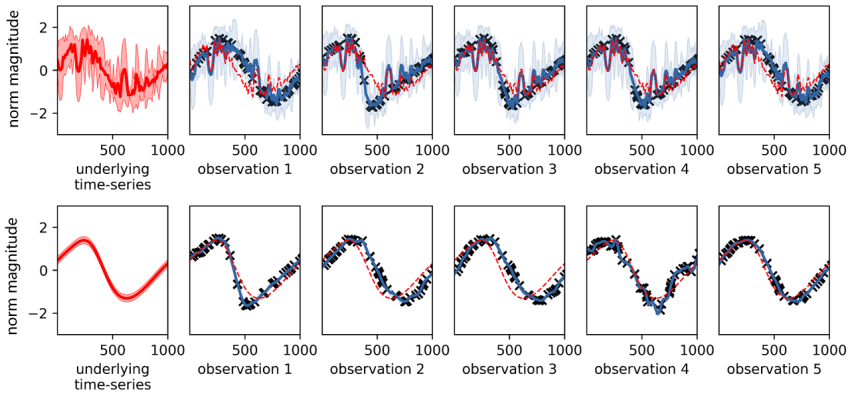


Figure 5: The result when using a periodic Matern 3/2 base kernel. In the upper figure note that the most likely solution is to fit to the local fluctuations, rather than the larger single period. In the lower figure, the correct period has been found, although this result requires many optimization restarts.

3.4. Anomaly scores

The process of evaluating the anomaly score requires us to compute the degree to which all points in the test data conform to the posterior model of star light-curves. The log-marginal likelihood can be used to achieve this.

For one time-series ($\mathbf{x}^*$, $\mathbf{y}^*$) the marginal likelihood is [7]:

$$\log p(\mathbf{y}^*|\mathbf{x}^*, \theta) = \log \int p(\mathbf{y}^*|\mathbf{f}, \mathbf{x}^*) p(\mathbf{f}|\mathbf{x}^*) d\mathbf{f} \tag{12}$$

$$= -\frac{1}{2}\log|\mathbf{K}_{\mathbf{x}^*\mathbf{x}^*}| - \frac{1}{2}(\mathbf{y}^* - \boldsymbol{\mu}_{\mathbf{x}^*})^\top \mathbf{K}^{-1}_{\mathbf{x}^*\mathbf{x}^*}(\mathbf{x}^* - \boldsymbol{\mu}_{\mathbf{x}^*}) - \frac{T}{2}\log(2\pi) \tag{13}$$

where $\mathbf{f}$ is the latent distribution over hierarchical functions, the covariance parameters (θ) have been inferred from training data and $\boldsymbol{\mu}_{\mathbf{x}^*}$ and $\mathbf{K}_{\mathbf{x}^*,\mathbf{x}^*}$ follow the definitions from Section 3.1. A natural anomaly score, is then the negative marginal likelihood since this will produce larger scores for non-conforming curves, *i.e.*

$$S(\mathbf{y}^*) = -\log\, p(\mathbf{y}^*|\mathbf{x}^*, \theta) \tag{14}$$

We can now infer the HGP model, and use the anomaly score in evaluating the degree to which new instances conform to the hierarchical distribution. We refer to this method of detecting (and ranking) anomalies as Hierarchical Gaussian Process Anomaly Detection (HGPAD).

4. Experimental Setup

4.1. Experiment data

We consider two datasets in our empirical work: MALLAT and OGLE. In this section, we will analyze the these datasets and also describe two baseline methods that HGPAD

is compared against. All the data sets we used in this project are downloaded from UCR time-series Classification Archive [47].

4.1.1. Non light-curve time-series data

MALLAT is a synthetic data set generated by Mallat in 1999 for the research of wavelets in signal processing [48]. This data set consists of eight classes, and there are 300 examples for each class. Each example has 1024 time points.

As we explained in Section 1 anomaly detection is not typically framed as a supervised learning problem. Hence, we design a 'known' anomalies evaluation approach to test our model. One outlier class and a few of normal classes should be defined. In the MALLAT dataset, Figure 6 shows three examples for class 3, 6 and 7 respectively.

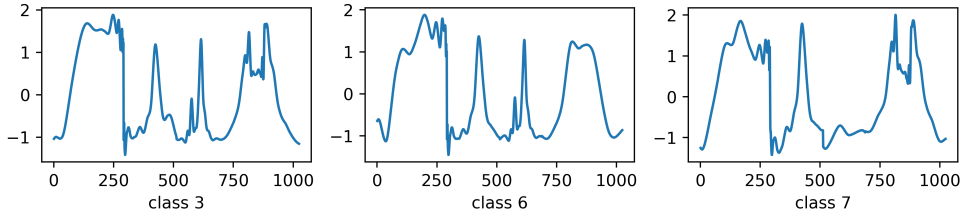


Figure 6: Three example curves from the synthetically generated MALLAT dataset. Each curve is from a different class; class 3 (left) class 6 (middle) and class 7 (right). These are synthetic timeseries data (the y-axis is on an aribtrary scale), but we can see that classes 3 and 6 are more similar to one another than to class 7 owing to the presence of two peaks at approx $x = 600$.

The shapes of class 3 and 6 are approximately similar except some little minor fluctuations on the rightmost peak and the shape of class 7 misses a peak in the middle compared with class 3 and 6. Thus, class 3 and 6 can be regarded as normal classes, and class 7 is the outlier class. This classification will be used in our experiments of the known anomalies detection.

4.1.2. Light-curve time-series data

Our specific research target is to find anomalies in light-curve time-series data called the Optical Gravitational Lensing Experiment (OGLE) [4], and is available at UCR Time-Series Classification Archive [47, 5].

The entire OGLE dataset contains 9236 light-curves, and each light-curve has 1024 time points. OGLE is class-labeled data. The curves are produced by three types of periodic variable stars: 1329 Cepheid (CEPH), 2580 Eclipsing Binaries (EB) and 5326 RR Lyrae (RRL). The light-curves of each star include an unknown number of anomalies.

Note that although CEPH and RRL arise from different star types, their shapes of light-curves are quite similar (see Figure 1). Hence, it is difficult to distinguish between CEPH and RRL because of this similarity. We can treat EB as the outlier class, and try to distinguish EB from CEPH and RRL. In this way, it can prove that our anomaly detection method is capable of discriminating those 'known' anomalies (EB) from two other, mutually similar classes (CEPH and RRL).

4.2. Baseline methods

As well as studying the empirical results of HGPAD, we also consider two baseline methods in our empirical evaluation which we discuss in the subsequent subsections.

4.2.1. Periodic Curve Anomaly Detection

Periodic Curve Anomaly Detection (PCAD) was proposed by Rebbapragada et al. to cope with the problem of anomaly detection on unsynchronized periodic time-series data [2]. PCAD is a kind of K-means clustering using cross correlation as a distance metric. OGLE, the dataset we used, is synchronized periodic time-series data because all the data have same begin and end time. Hence, a simple version PCAD was implemented that did not incorporate an updating phase. Algorithm 1 shows the pseudocode of this (using the notation given in [2]). The inner loop of this algorithm has three main stages: calculation of distances between instances and centroids, calculation of cluster assignments, and re-calculation of centroids. Distance calculation here is based on the maximal cross-correlation between the centroids and examples, and instances are then assigned to the closest cluster based on this. Centroids are then re-estimated via the mean of instances assigned to the cluster.

The convergence requirement is that the quantization error must be non-increasing. In this case, quantization error can be denoted by:

$$\|\mathbf{y}^* - \mathbf{c}_k\|_2^2 \tag{15}$$

where $\|\cdot\|_2^2$ represents the squared ℓ_2 norm of a vector, $\mathbf{y}^*$ is an arbitrary instance and $\mathbf{c}_k$ is the k-th centroid of PCAD. With K centroids (where K is selected with the Bayesian Information Criteria (BIC) [49] on validation data), the anomaly scores can be calculated as follows:

$$score(\mathbf{y}^*) = \sum_{k=1}^{K} \frac{|n_k|}{n} r^2_{\mathbf{y}^*,\mathbf{c}_k} \tag{16}$$

where $|n_k|$ is the number of time-series whose closes centroid is $\mathbf{c}_k$ (this value will have been calculated during training and is stored by the model), n indicates the size of the dataset and $r^2_{\mathbf{y}^*,\mathbf{c}_k}$ means the square of distance between instance $\mathbf{y}^*$ and the k-th centroid by their cross correlation.

Notice that some implementation details are missing in [2], for example, the maximum number of iterations if the quantization error keeps changing. Hence, we simplify PCAD and add another limitation (maximum number of iteration) to the convergence requirement.

Algorithm 1 PCAD algorithm

function PCAD($x[], k$) ▷ time-series dataset x, number of centroids k
  initialize $centroids[]$ ▷ randomly select according to k
  while not converged **do**
    $bestcentroids[] \leftarrow calculateDistance(x[], centroids[])$ ▷ uses cross-correlation as distance metric
    $cluster[][] \leftarrow assignClusters(x[], bestcentroids[])$
    $centroids[] \leftarrow recalculateCentroids(x[], cluster[][])$
  end while
  return $centroids[]$ ▷ trained $centroids[]$ will be used for calculating anomaly score
end function

4.2.2. RAND-C

As discussed in [2], not many published alternative methods to PCAD are available (particularly when considering its ability to generalise to large datasets). Several baseline methods were proposed by the authors. We elected to use their random centroids (RAND-C) baseline method in this work since it performs well in their evaluation. Other baselines (including PCAD with $K = 1$) were not selected owing to their poor performance in the evaluation on the OGLE and MALLAT datasets in [2].

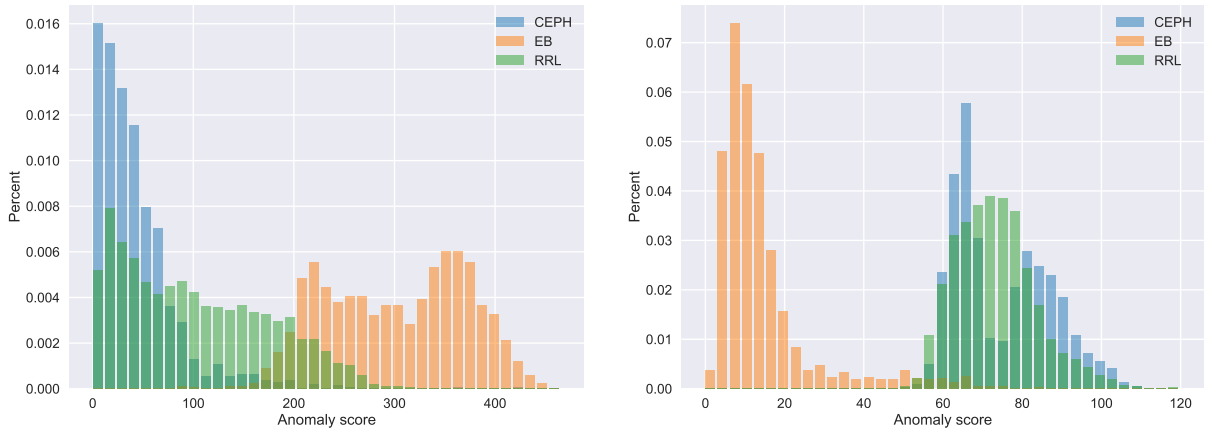
In RAND-C, K centroids are uniformly sampled from the dataset in order to produce a reference set of centroids which, in contrast to PCAD, remain fixed and are not updated. The basic approach of PCAD is used by RAND-C, namely cross-correlation forms the distance measure between centroids and instances in scoring instance. A practitioner is still required to specify K, and this is selected by cross-validation with BIC. Although the centroid locations are not optimised in this method, it is shown to be a strong baseline in [2].

5. Results

In this section we outline the main experimental results of our work. First, we consider the utility of 'known anomaly detection'. This is a setting where the ground truth labels are used to stratify inlying and outlying sequences (*i.e.* classes 3 and 6 are inliers and class 7 is the outlier with MALLAT). Performance accuracy can easily be evaluated in this experiment since ground truth labels are available. In these experiments we also consider the quantity of data required to produce stable models. Subsequently, we present a study of 'unknown anomalies'. This is a setting where one star type, *e.g.* CEPH, is considered and is used to rank all CEPH instances by anomaly score. We then visually assess the instances with largest and smallest anomaly scores.

5.1. Detection of known anomalies

Figure 7a shows the distribution over scores for each class for a HGPAD trained on the CEPH class, and Figure 7b, shows the anomaly score for the model when trained on the EB class. We can quantify the ability of our model to detect anomalous data by performing inference with HGPAD on one class, and testing the anomaly score on all other classes.



(a) CEPH (b) EB/RRL

Figure 7: This figure depicts the histogram of anomaly scores in two training scenarios. Figure 7a shows the anomaly score when trained on CEPH. Similarly Figure 7b shows the anomaly score when trained on the EB and RRL classes.

In this experiment we apply HGPAD to infer an underlying function for MALLAT dataset by using its different proportions of the inliers as training instances, then construct a mixed test set including 10% outliers (class 7) and 90% normal instances (classes 3 and 6) to test the precision of our model. We treat the OGLE dataset in a similar manner, where we use CEPH and RRL as the inlier classes and EB as the outlier class. Since we know the number of anomalies in the testing dataset (n), the precision can be calculated by looking at the top n instances in the output of the HGPAD model.

Table 1: Predictive precision for OGLE and MALLAT dataset.

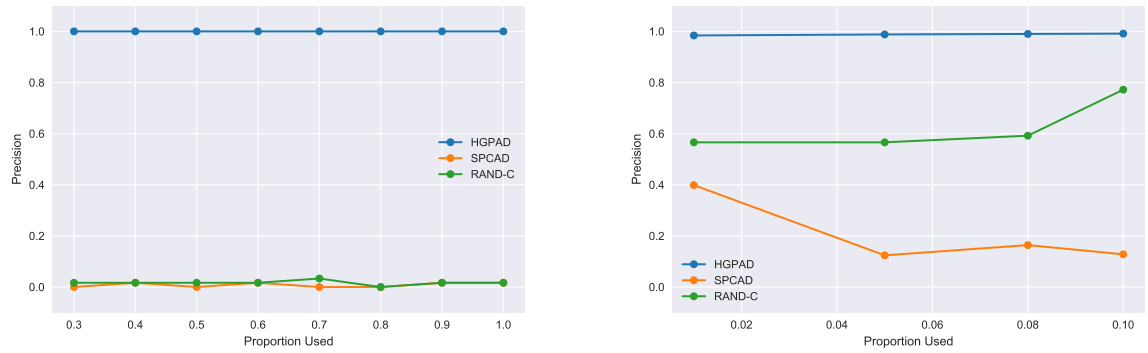
	RAND-C	SPCAD	HGPAD
MALLAT	0.02	0.02	1.00
OGLE	0.44	0.16	0.99

We summarise the accuracy of the models in Table 1, where we can see that the proposed HGPAD model achieves the best predictive performance over all baselines on both datasets considered. It is important to mention that the performance difference between HGPAD and the baseline methods (both here and in later results) is significant; *e.g.* in the case of the MALLAT dataset the precision difference is ≈ 0.98 and with the OGLE dataset HGPAD outperforms the baseline methods by 0.56 and 0.84 as also reported in [44]. We are not yet in a position to fully characterise the underlying cause of this but are able to outline some possible explanations. We note that in the experimental analysis of [2] a large discrepancy is also reported between their proposed and baseline approaches. This leads us to believe that the dataset here may, in some sense, be 'unstable', *i.e.* the small differences in the data lead to a significant change in predictions. Another complementary perspective may be that

PCAD has overfitted the training data with the value K. Although this was selected with cross-validation, the selection criterion (BIC) has historically been criticised for use in model selection [50] and we did not explore alternative parameter selection techniques. In contrast, we have already discussed how HPGAD is very stable in terms of convergence to good posterior models in Section 3.3. This stability derives from its consideration of covariance between all instance pairs as opposed to instance-to-centroid correlations in PCAD and RAND-C. It is worth noting that while a simplified version of PCAD has been used in our work we also experimented with the full algorithm outlined in [2] and achieved similar baseline results to those presented in Table 1. A final possible explanation for the difference in performance is that in PCAD the mean of cluster assignments is used to re-estimate the centroid position. Since the distance measure is the cross-correlation (as opposed to Euclidean distance) it is possible that the mean is ill-suited for re-estimating the centroid.

5.2. Effect of Dataset Size

Of paramount interest to us is the effect of dataset size on the detection of anomalies and we investigate the effect of dataset size to the robustness of anomaly detection in our HGPAD and baseline models. We show the results in Table 2 and Figure 8 for the MALLAT and OGLE datasets. We can see that the proposed model consistently out-performs the baseline models in terms of precision, even when using small fractions of the total dataset. We believe that the implicit uncertainty quantification of the HGPAD model contributes to this since it reduces the likelihood of overfitting.



(a) MALLAT (b) OGLE

Figure 8: Precision of anomaly detection with known anomalies as a function of the training sample size for MALLAT (left) and OGLE (right).

We consider maximally only 10% of the OGLE data due to the high computational cost of the methods. On the other hand, we can see that by utilising only a small subset of the available data our powerful model maintains a high level of accuracy. It is well worth noting that when training from the OGLE dataset, there are only 725 instances availalble and by using just 1% of these we achieve very good anomaly detection by using HGPAD. There has been a slight rise in the precision of RAND-C in OGLE from 56% to 77%. However, the

Table 2: Precision for MALLAT & OGLE for increasing dataset sizes.

	MALLAT			OGLE		
Data size	0.4	0.7	1.0	0.01	0.08	0.1
RAND-C	0.01	0.03	0.01	0.57	0.59	0.77
SPCAD	0.01	0.01	0.01	0.40	0.16	0.12
HGPAD	1.00	1.00	1.00	0.98	0.99	0.99

precision of another distance-based approach, SPCAD, has fallen from around 40% to 13%. Meanwhile, both SPCAD and RAND-C almost totally do not work in detecting anomalies in MALLAT dataset. As mentioned before, the not significant dissimilarity of the shapes of the outlier and normal classes in MALLAT may cause that distance-based methods (SPCAD and RAND-C) do not work well. To sum up, HGPAD is the most stable one among these three anomaly detection methods even if only a small set of data is used to train.

5.3. Effect of Training with Anomalies

In this experiment we make the assumption that the precision of the star light-curve classifier will be contaminated by the presence of anomalies in the training set, and we present a method to remove these from consideration in the pipeline. To achieve this we assume that we have access to a labelled dataset. The dataset is partitioned into three subsets: training, validation and test. We use a relatively small training set (approx 1% as motivated in the results from Section 5.2) to learn HGPAD models. We then deploy these on the validation set and sort the instances according to their anomaly score. Then, Q% of validation instances with the lowest anomaly score (*i.e.* the least anomalous instances) are selected to learn a second HGPAD model. This, then, is used to estimate the precision of star light-curve prediction on the test set according to the methodology of the previous section. Intuitively, if anomalies are present in the training data, as Q approaches 100% we will certainly have trained the prediction model with anomalous data, and thus may be more susceptible to misclassification error. However, by chaining the anomaly detection as above we train only on 'well behaved' curve examples. We discuss the effect of this on the precision estimation on the test set below.

Figure 9 presents the results of this experiment. We can see that by keeping only a small quantity of the data models with high precision of outlier detection are produced. However, as we increase the retention percentage above 80% we can see that the performance of outlier detection degrades as expected. It is difficult to interpret this image fully, however we can see that, as expected, the precision of anomaly detection degrades as anomalies are introduced to the HGPAD model. This is because the outlier class now receives scores comparable to the inlier class owing to the presence of outliers in the training data, and HGPAD has therefore become a less effective ranker of anomalous examples. We show examples of the rejected light-curves for all star types in the next section.

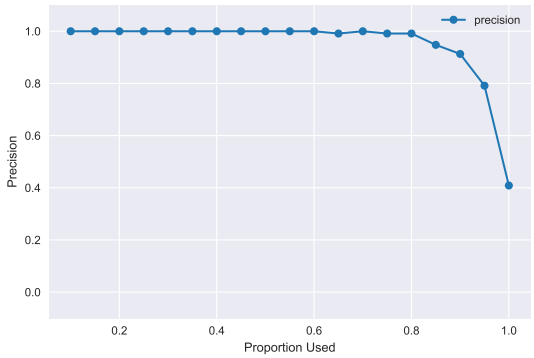


Figure 9: The impurity of the testing dataset has an influence on the precision.

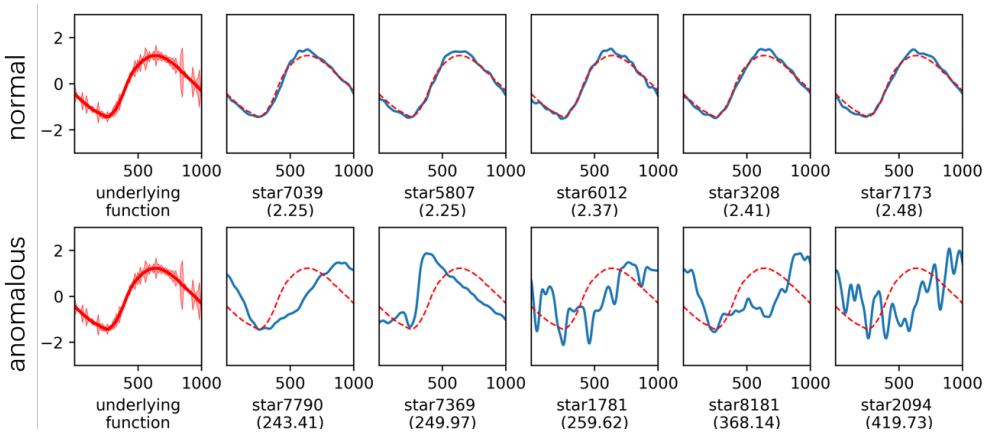
5.4. Detection of unknown anomalies

It is impossible to report precision for an anomaly detection model if we do not have any knowledge of the number of anomalies in the testing dataset. For the analysis of unknown anomalies within each class of light-curves, we first infer a HGPAD model on a particular light-curve class and compute anomaly scores. We then rank the held out dataset from least anomalous to most anomalous. We illustrate this idea in Figures 10a to 10c for CEPH, EB, and RRL light-curves respectively. In each figure, the leftmost column depicts the underlying latent light-curve function that was inferred by HGPAD. The top row depicts the 5 light-curves that received the smallest anomaly score. We can see that there is a close match between the latent function and these instances in this figure. The bottom row of each sub-figure shows the instances that received the highest anomaly score. We can see that the latter light-curves do not closely resemble any of the prototypical curves, and hence are likely to be actual outliers. These could then be checked by an expert and either assigned to one of the other existing classes or classified as an entirely new class of star.

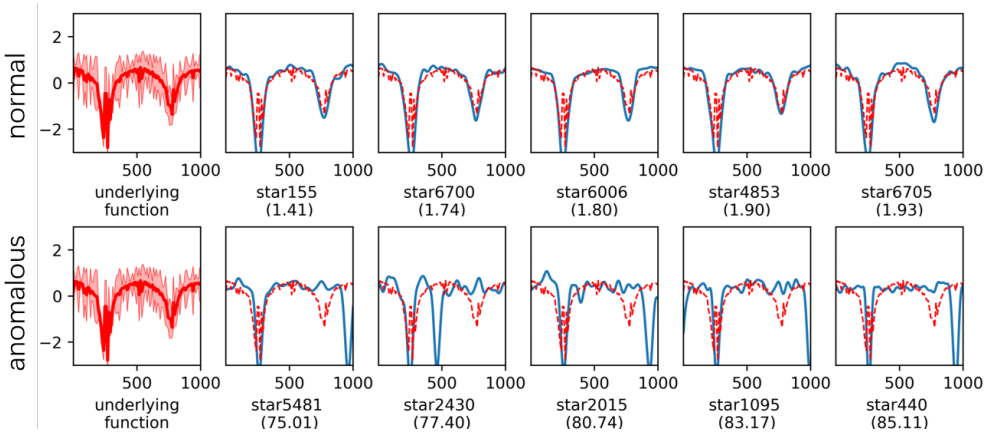
6. Conclusions

In this paper hierarchical Gaussian processes are introduced for anomaly detection in astronomical timeseries data. The proposed approach is shown to out-perform two baseline methods over two datasets, both in terms of detecting and ranking anomalous time-series in labelled experiments and in terms of visual analytics in completely unsupervised scenarios. While we incur several additional parameters, the probabilistic programming framework will optimise these during inference. We show experimentally that the proposed approach is able to adapt to different situations and consistently outperforms baseline methods.

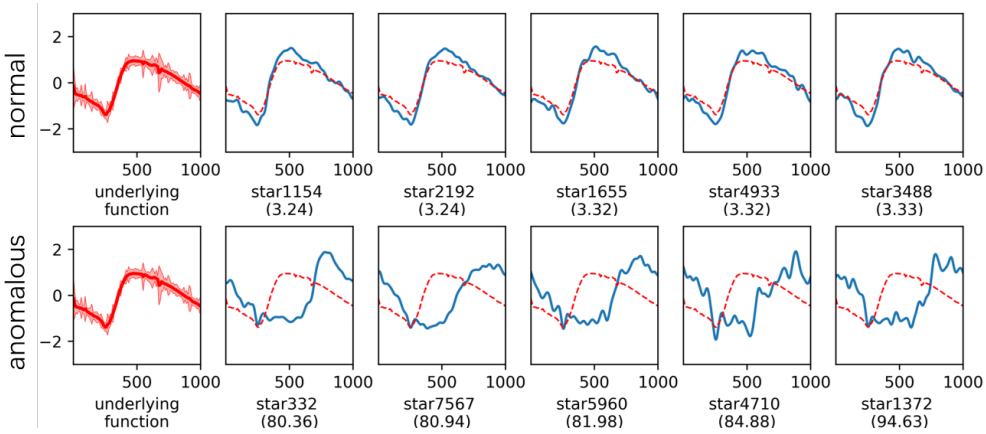
A key advantage of hierarchical Gaussian processes for anomaly detection is that near-optimal outlier detection is achieved even when using only a small numbers of instances. In particular we show in this paper that our approach identifies anomalies with almost perfect precision with only seven labelled instances to infer the model. This can bring several practical benefits, in particular financially since the acquisition of labelled instances



(a) CEPH



(b) EB



(c) RRL

Figure 10: Normal (odd rows) and anomalous (even rows) star light-curves.

is often very expensive. We also illustrate how the presence of anomalies can contaminate the utility of outlier detection models, and present an iterative filtering approach for reducing the number of outliers used in generating the anomaly detection model. An interesting side-effect of this approach is that it appears to suggest the proportion of outliers in the dataset.

We have been inspired by the generalisability of the proposed method from small datasets and will consider marrying online and active learning methods to the methodology in future work. We will also explore techniques for reducing the computational cost that is associated with the approach (*e.g.* low-rank approximations to the covariance matrix or using inducing-points [51, 52]) and this will enable application to larger datasets and different application areas, such as detecting financial fraud, detecting anomalous heart rate records, etc.

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 15.

[2] U. Rebbapragada, P. Protopapas, C. E. Brodley, C. Alcock, Finding anomalous periodic time series, Machine Learning 74 (3) (2008) 281–313. doi:10.1007/s10994-008-5093-3.

[3] P. B. Stetson, On the automatic determination of light-curve parameters for Cepheid variables, Publications of the Astronomical Society of the Pacific 108 (728) (1996) 851.

[4] OGLE, http://ogle.astrouw.edu.pl/.

[5] D. Yankov, et al., Disk aware discord discovery: Finding unusual time series in terabyte sized datasets, Knowledge and Information Systems 17 (2) (2008) 241–262.

[6] C. Sterken, C. Jaschek, Light curves of variable stars: a pictorial atlas, Cambridge University Press, 2005.

[7] C. E. Rasmussen, C. K. Williams, Gaussian processes in machine learning, Lecture notes in computer science 3176 (2004) 63–71.

[8] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, S. Aigrain, Gaussian processes for time-series modelling, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 371 (1984). doi:10.1098/rsta.2011.0550.

[9] J. Hensman, et al., Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters, BMC bioinformatics 14 (1) (2013) 252.

[10] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, NY, USA, 2012, pp. 262–270. doi:10.1145/2339530.2339576.
URL http://doi.acm.org/10.1145/2339530.2339576

[11] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: Proceedings, Presses universitaires de Louvain, 2015, p. 89.

[12] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, S. Frankenstein, M. Lerner, Grammarviz 2.0: a tool for grammar-based pattern discovery in time series, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 468–472.

[13] G. Ratsch, S. Mika, B. Scholkopf, K.-R. Muller, Constructing boosting algorithms from svms: an application to one-class classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (9) (2002) 1184–1199.

[14] S. Aghabozorgi, A. S. Shirkhorshidi, T. Y. Wah, Time-series clustering–a decade review, Information Systems 53 (2015) 16–38.

[15] M. Gupta, J. Gao, C. C. Aggarwal, J. Han, Outlier detection for temporal data: A survey, IEEE Transactions on Knowledge and Data Engineering 26 (9) (2014) 2250–2267.

[16] P. Domingos, The role of occam's razor in knowledge discovery, Data mining and knowledge discovery 3 (4) (1999) 409–425.
[17] I. J. Myung, M. A. Pitt, Applying occam's razor in modeling cognition: A bayesian approach, Psychonomic Bulletin & Review 4 (1) (1997) 79–95.
[18] C. E. Rasmussen, Z. Ghahramani, Occam's razor, in: Advances in neural information processing systems, 2001, pp. 294–300.
[19] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.
[20] K. P. Murphy, Machine learning : a probabilistic perspective, 1st Edition, MIT Press, 2013.
URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262018020
[21] S. J. Roberts, Novelty detection using extreme value statistics, IEE Proceedings-Vision, Image and Signal Processing 146 (3) (1999) 124–129.
[22] N. Twomey, A. Temko, J. Hourihane, W. P. Marnane, Automated detection of perturbed cardiac physiology during oral food allergen challenge in children, IEEE journal of biomedical and health informatics 18 (3) (2014) 1051–1057.
[23] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1975–1981.
[24] H. N. Akouemo, R. J. Povinelli, Probabilistic anomaly detection in natural gas time series data, International Journal of Forecasting 32 (3) (2016) 948–956.
[25] T. Xiang, S. Gong, Video behavior profiling for anomaly detection, IEEE transactions on pattern analysis and machine intelligence 30 (5) (2008) 893–908.
[26] J. Winn, C. M. Bishop, T. R. Diethe, Model-Based Machine Learning, Microsoft Research, 2015.
[27] R. Herbrich, T. Minka, T. Graepel, Trueskill: a bayesian skill rating system, in: Advances in neural information processing systems, 2007, pp. 569–576.
[28] P. A. Flach, S. Spiegler, B. Golénia, S. Price, J. Guiver, R. Herbrich, T. Graepel, M. J. Zaki, Novel tools to streamline the conference review process: experiences from sigkdd'09, ACM SIGKDD Explorations Newsletter 11 (2) (2010) 63–67.
[29] D. H. Stern, R. Herbrich, T. Graepel, Matchbox: large scale online bayesian recommendations, in: Proceedings of the 18th international conference on World wide web, ACM, 2009, pp. 111–120.
[30] D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, D. M. Blei, Edward: A library for probabilistic modeling, inference, and criticism, arXiv preprint arXiv:1610.09787.
[31] A. Patil, D. Huard, C. J. Fonnesbeck, Pymc: Bayesian stochastic modelling in python, Journal of statistical software 35 (4) (2010) 1.
[32] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, A. Riddell, Stan: A probabilistic programming language, Journal of statistical software 76 (1).
[33] Z. Dai, E. Meissner, N. D. Lawrence, MXFusion: A modular deep probabilistic programming library, in: NIPS 2018 Workshop on Machine Learning Open Source Software, 2018.
[34] S. Mascaro, A. E. Nicholso, K. B. Korb, Anomaly detection in vessel tracks using bayesian networks, International Journal of Approximate Reasoning 55 (1) (2014) 84–98.
[35] N. Ye, et al., A markov chain model of temporal behavior for anomaly detection, in: Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, Vol. 166, West Point, NY, 2000, p. 169.
[36] R. Darkins, E. J. Cooke, Z. Ghahramani, P. D. Kirk, D. L. Wild, R. S. Savage, Accelerating bayesian hierarchical clustering of time series data with a randomised algorithm, PloS one 8 (4) (2013) e59795.
[37] E. Yu, P. Parekh, A bayesian ensemble for unsupervised anomaly detection, arXiv preprint arXiv:1610.07677.
[38] R. M. Neal, Markov chain sampling methods for dirichlet process mixture models, Journal of computational and graphical statistics 9 (2) (2000) 249–265.
[39] J. Van Gael, A. Vlachos, Z. Ghahramani, The infinite hmm for unsupervised pos tagging, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume

2, Association for Computational Linguistics, 2009, pp. 678–687.
[40] GPy, GPy: A gaussian process framework in python, http://github.com/SheffieldML/GPy (since 2012).
[41] D. G. Matthews, G. Alexander, M. Van Der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, J. Hensman, Gpflow: A gaussian process library using tensorflow, The Journal of Machine Learning Research 18 (1) (2017) 1299–1304.
[42] S. Reece, R. Garnett, M. Osborne, S. Roberts, Anomaly detection and removal using non-stationary Gaussian processes, arXiv preprint arXiv:1507.00566.
[43] W. Herlands, E. McFowland III, A. G. Wilson, D. B. Neill, Gaussian process subset scanning for anomalous pattern detection in non-iid data, arXiv preprint arXiv:1804.01466.
[44] H. Chen, T. Diethe, N. Twomey, P. Flach, Anomaly detection in star light curves using hierarchical Gaussian processes, in: Proceedings of the Twenty-sixth European Symposium on Artificial Neural Networks, 2018, pp. 615–620.
[45] D. Duvenaud, Automatic model construction with Gaussian processes, Ph.D. thesis, University of Cambridge (2014).
[46] The Kernel Cookbook, http://www.cs.toronto.edu/~duvenaud/cookbook/index.html.
[47] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR time series classification archive, www.cs.ucr.edu/~eamonn/time_series_data/ (July 2015).
[48] S. Mallat, A wavelet tour of signal processing, Academic press, 1999.
[49] D. Pelleg, A. W. Moore, et al., X-means: Extending k-means with efficient estimation of the number of clusters., in: Icml, Vol. 1, 2000, pp. 727–734.
[50] D. L. Weakliem, A critique of the bayesian information criterion for model selection, Sociological Methods & Research 27 (3) (1999) 359–397.
[51] J. Quiñonero-Candela, C. E. Rasmussen, A unifying view of sparse approximate gaussian process regression, Journal of Machine Learning Research 6 (Dec) (2005) 1939–1959.
[52] E. Snelson, Z. Ghahramani, Sparse gaussian processes using pseudo-inputs, in: Advances in neural information processing systems, 2006, pp. 1257–1264.